Chapter

2

Azure Core Services

MICROSOFT EXAM OBJECTIVES COVERED IN THIS CHAPTER:

DESCRIBE CORE AZURE SERVICES

✓ **Describe the core Azure architectural components**

- Describe the benefits and usage of Regions and Region Pairs
- Describe the benefits and usage of Availability Zones
- Describe the benefits and usage of Resource Groups
- Describe the benefits and usage of Subscriptions
- Describe the benefits and usage of Management Groups
- Describe the benefits and usage of Azure Resource Manager
- Explain Azure resources

DESCRIBE CORE AZURE SERVICES

✓ **Describe core resources available in Azure**

- Describe the benefits and usage of Virtual Machines, Azure App Services, Azure Container Instances (ACI), Azure Kubernetes Service (AKS), and Windows Virtual Desktop
- Describe the benefits and usage of Container (Blob) Storage, Disk Storage, File Storage, and storage tiers
- Describe the benefits and usage of Cosmos DB, Azure SQL Database, Azure Database for MySQL, Azure Database for PostgreSQL, and SQL Managed Instance
- Describe the benefits and usage of Azure Marketplace

DESCRIBE CORE SOLUTIONS AND MANAGEMENT TOOLS ON AZURE

✓ **Describe Azure management tools**

- Describe the functionality and usage of Azure Resource Manager (ARM) templates

Microsoft Azure consists of a multitude of services. Even though the number of different Azure services can be bewildering, there are nevertheless core services and concepts that underpin many of the Azure offerings. Understanding the role that these core services play is instrumental in understanding Azure holistically. The exam objectives explored in this chapter are designed to help you build that understanding.

First, let's explore concepts that will help you understand where and how Azure services are deployed geographically, how high availability works, and ways to manage those services.

The AZ-900 exam covers several network services that are core services supporting almost every Azure service in some way. These networking services and concepts are covered in Chapter 3, "Azure Core Networking Services." Azure solutions and management tools are covered in Chapter 5, "Azure Solutions."

Core Azure Architectural Components

Azure is a distributed cloud offering with data centers located in many geographical regions across the world. That global distribution primarily provides a means of serving customers close to where their users are; enables customers to meet legal, compliance, or tax requirements; and enhances opportunities for high availability. The following sections explain these core concepts by examining the roles that regions, availability zones, resource groups, and Azure Resource Manager play in enabling and supporting that distributed cloud model.

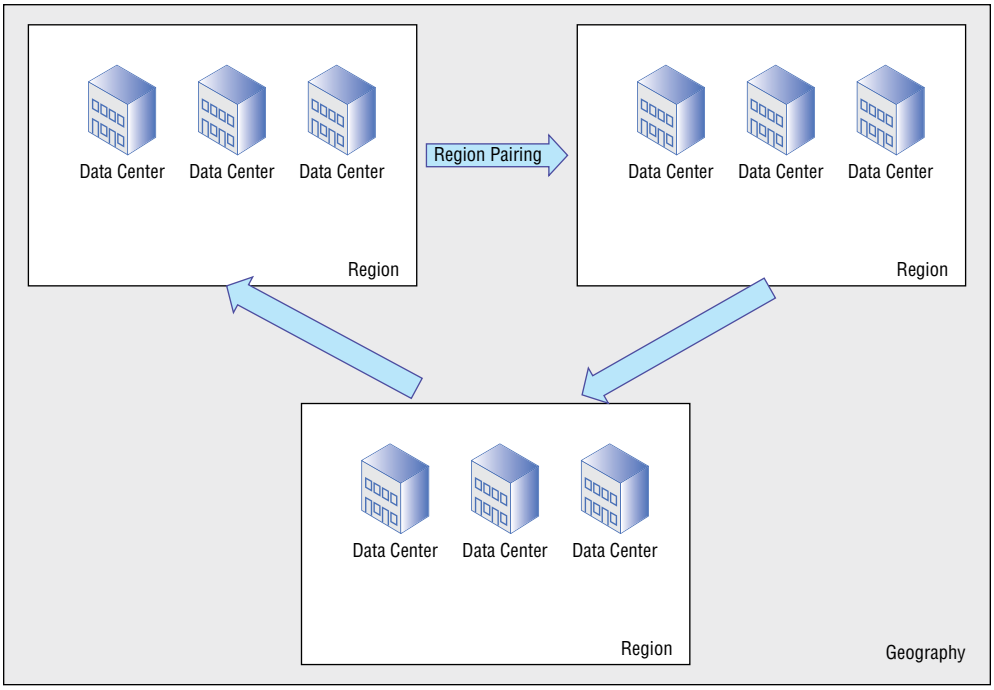
Geographies and Regions

As explained previously in this chapter, Azure has multiple data centers distributed around the globe. Those data centers house the servers and other infrastructure on which Azure is built. The data centers are distributed into various *geographies*. The Azure geographies generally align to specific countries such as the United States, Canada, Australia, and so on. However, Azure geographies can also be aligned to specific markets, such as Europe and Asia.

Compliance and data residency are key aspects of an Azure geography, and China is a good example. Azure China is a physically isolated instance of Azure located wholly in China. Azure China is operated by Shanghai Blue Cloud Technology Co., Ltd. (21Vianet). Azure China enables Chinese companies and entities to host their data and applications within China and meet strict Chinese regulatory requirements. Azure China is not limited to Chinese government entities.

Within each geography are Azure *regions*. A region is a grouping of data centers that interact to provide redundancy and availability for the services hosted within that region. For example, West US, Central US, and North Central US are three of many regions in the United States. Each region is paired with another in the same geography to allow for replication of resources across multiple data centers to reduce the effects of natural disasters, outages, or other potential events that would affect a given data center's ability to serve up the services hosted in that data center. For example, West US and East US are paired regions, North Europe and West Europe are paired, and UK West and UK South are paired. Figure 2.1 shows the relationship between geographies, region pairs, regions, and data centers.

FIGURE 2.1 The relationships between geographies, regions, region pairs, and data centers



Microsoft establishes and controls the pairing relationships between regions, which means that you cannot choose a region pair. However, you choose the region in which to deploy a service, which indirectly determines which other region is in the pair.

You are not limited to choosing a single region to deploy your Azure services, nor are you limited to a specific region pair. In fact, you don't have to use a regional pair at all and can instead host all your services in a single region. However, using region pairs offers many benefits for redundancy, resiliency, and business continuity.

Visit `https://azure.microsoft.com/en-us/global-infrastructure/geographies` to learn more about Azure geographies and regions, and to view information about region location and pairing, which will help you choose the right region(s) to host your services in Azure.

Availability Zones

As Figure 2.1 illustrates, there is a nested nature to Azure that provides both fault tolerance and availability. Azure offers another level of availability protection through *availability zones*. An availability zone is a physically separate zone within a region, each with its own power, network, and cooling. You might think of an availability zone as a data center, although the separation of power, network, and cooling defines the zone, not the physical data center. An availability zone might encompass more than one data center. There are a minimum of three availability zones per region, although not all regions offer availability zones. Figure 2.2 illustrates availability zones.

For example, assume you need to deploy a set of virtual machines (VMs) to host a line-of-business service but need to ensure that the service remains available in the event of a failure at one of the data centers hosting the VMs. You deploy VMs to an additional availability zone so that if an incident does occur at one of the data centers, the VMs in the other availability zone will be unaffected.

Each Azure availability zone encompasses a fault domain and an update domain. So, if you deploy VMs across three availability zones, those VMs will not be updated at the same time because they are located in three different update domains.

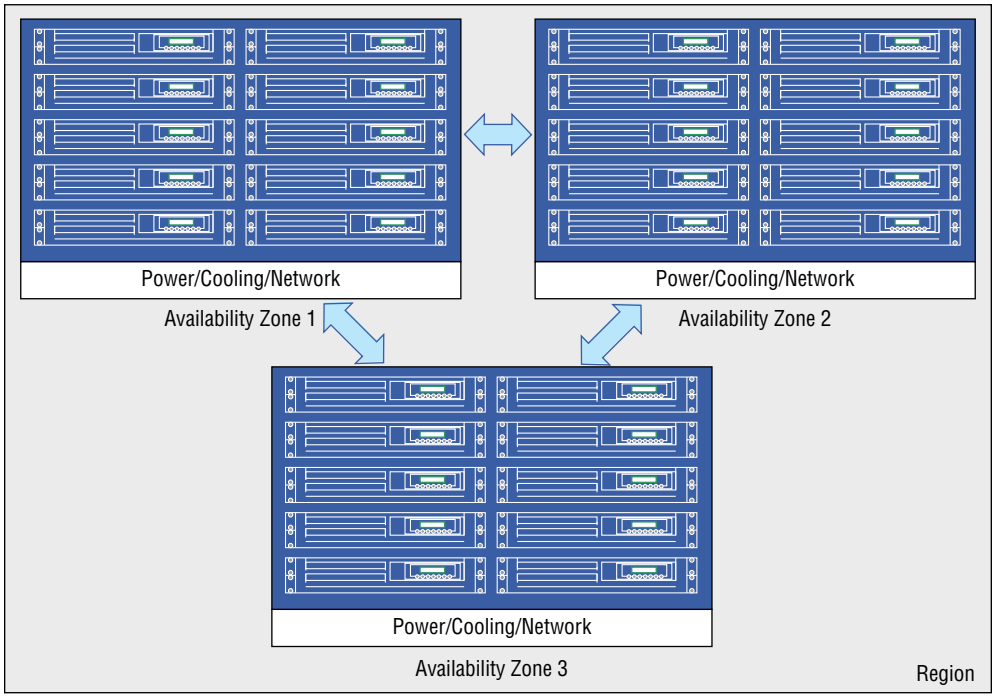
Azure includes two categories of services that support availability zones:

- **Zonal services:** Resources are pinned to a specific zone. To ensure redundancy for zonal services, you must deploy the services across multiple availability zones. Virtual machines are an example of a zonal service. To take advantage of availability zones with VMs, you must deploy them accordingly.

- **Zone-redundant services:** Azure replicates the service automatically across zones. Storage is a good example. When you deploy a new storage account, choosing ZRS (Zone Redundant Storage) will result in the storage automatically being replicated across availability zones. SQL databases are another example of a zone-redundant service.

You must use availability zones if you want to achieve certain SLAs. For example, deploying VMs to two or more availability zones results in an SLA of 99.99 percent.

FIGURE 2.2 Availability zones offer an additional layer of service availability.



There is no additional cost for VMs deployed in an availability zone, although you will incur charges for VM-to-VM data transfers between zones.

Bringing It All Together

As the previous sections describe, Azure offers multiple levels of availability and fault protection. The following list summarizes each:

- **Geography:** A geography defines a discrete market that preserves data residency and compliance boundaries. Geographies typically contain two or more regions. Although often defined by country, Azure geographies are not constrained to a specific country.
- **Region:** A region is a collection of data centers that interact to provide data redundancy and service availability.
- **Region pair:** A region pair consists of two regions within the same geography. Updates are rolled out to regions serially, meaning only one region is updated at a time. The second region in a pair is updated only after the first is successfully updated. An issue with an update or an outage would then affect only one region in the pair. Some Azure services take advantage of region pairs to guard against data loss, replicating data between regions.
- **Data center:** Individual data centers within a region host the servers and other infrastructure needed to host services within that region.
- **Availability zones:** An availability zone encompasses separate power, networking, and cooling, and it is intended to guard against data loss or outages caused by failures in any of those three categories. Although a single data center generally fits those criteria, a data center is not an availability zone, and vice versa. Conceptually, however, they are much the same. Deploying services across availability zones enables you to achieve higher SLAs for those services.

In summary, each of the topics discussed in the previous sections provides some level of fault tolerance and availability. When you're developing an Azure strategy, one of your first considerations is to choose a geography that meets your compliance and regulatory requirements for data residency. The next consideration is the region(s) in which you will host your data, services, and applications. You might choose multiple regions depending on specific needs or how your users are geographically dispersed. For example, you might host some services in West US for users located on the West Coast of the United States, and East US for users located on the East Coast. Or, perhaps you host content in Central US, then host appropriate services or applications in East US and West US.

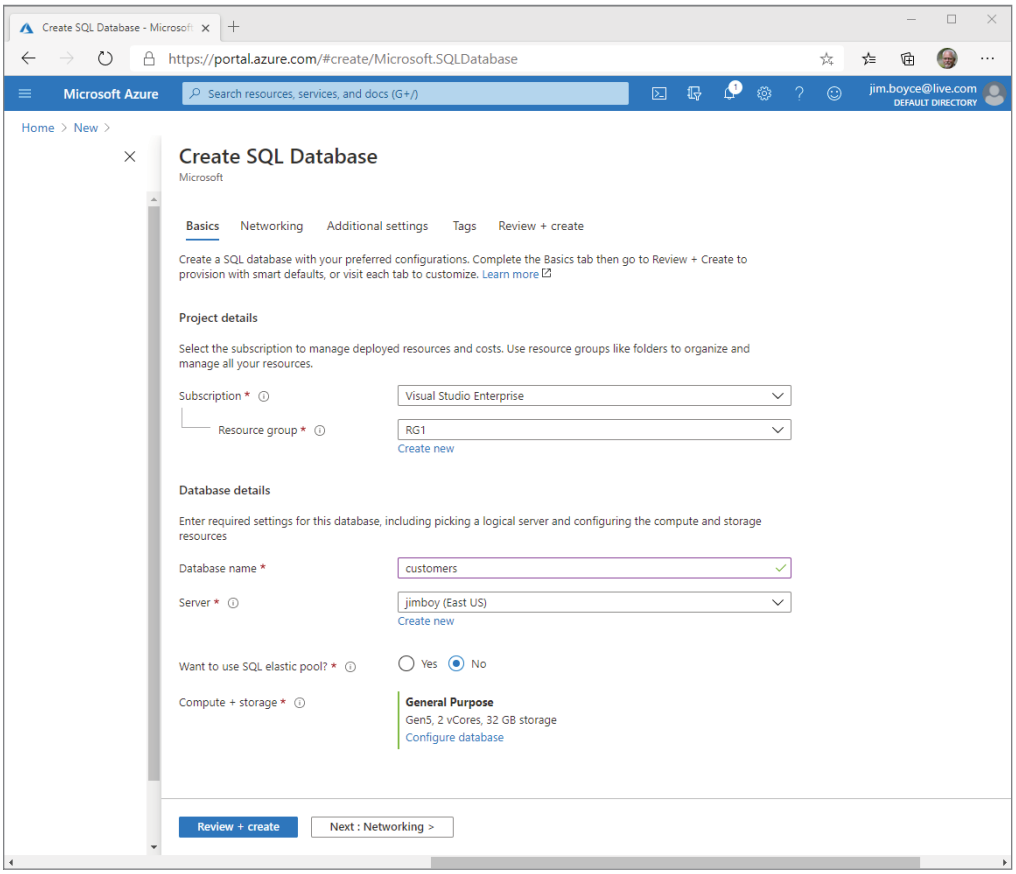
Once you've identified the regions you'll be using and how those regions are paired, you can begin making decisions regarding not only where you will host storage and other services, but what additional types of fault tolerance and availability are appropriate. That might take the form of replicating storage, data, or other services between regions, or it might mean using other services to provide fault tolerance, redundancy, and high availability. Selecting the appropriate geography and region(s) underpins all of those considerations.

For more details on regions and availability zones, see `https://docs.microsoft.com/en-us/azure/availability-zones/az-overview`.

Resources and Resource Groups

Azure *resources* are manageable items in Azure such as virtual machines, databases, virtual networks, and storage accounts. Figure 2.3 shows an example of creating a resource in Azure.

FIGURE 2.3 Creating a resource in Azure



If you deploy only a few resources to Azure, managing them will likely not be a significant challenge. However, your Azure solution may grow to encompass a very large number of resources of different types, all potentially scattered across multiple regions. As the number of resources grows, your ability to manage them individually quickly becomes difficult. *Resource groups* provide a means for organizing and managing resources.

Think of a resource group as a logical container for one or more resources. You can apply various properties to the resource group, and those properties apply to all the resources in that resource group. For example, assume you have a group of resources that you want to prevent from being deleted but want resource admins to be able to modify. You can create a resource group, assign the resources to that group, and apply a CanNotDelete lock on the resource group. Any resources that you later apply to the resource group will also be automatically protected from deletion.

Protecting resources from deletion is just one function of resource groups. You can use resource groups to apply policies to a group of resources, control who has administrative permissions for the resources in the group, and perform other operations on the resources as a single entity. For example, assume you stand up 10 VMs for a proof-of-concept project. When the project is completed and it's time to delete the VMs, you simply delete the resource group. All the VMs in the resource group, along with any supporting resources that you also added to the resource group for the project, are deleted.

The following list describes several points to consider when using resource groups and managing resources:

- **Lifecycle:** All resources in a resource group should share the same lifecycle for deployment, updates, and deletion.
- **Resource assignment:** A resource can exist in only one group, but you can add or remove a resource to or from the group as needed. You can also move resources from one group to another.
- **Location:** Resources in a group can reside in various regions (again, resource groups are *logical* containers that provide a means to apply various properties to resources they contain—it doesn't matter where the resources physically reside). However, resource groups do have an assigned location that determines where the group's metadata is stored. Compliance requirements can therefore dictate where you create your resource groups. Also, if the region where the resource group resides is unavailable, you will be unable to update the resources in the group because the metadata is unavailable. However, the resources can continue to function.
- **Scope:** Resource groups enable you to apply management scope to resources collectively. You can assign Azure policies, Azure roles, or resource locks to the group, which then apply to the resources contained within it.
- **Resource interaction:** Resources in different groups can interact with one another. For example, an application server in one resource group might use a database in a different resource group.
- **Deletion:** When you delete a resource group, all resources in the group are deleted.
- **Creation:** You can use the Azure portal, PowerShell, Azure CLI, or an Azure Resource Manager template to create a resource group.
- **Tags:** You can apply tags to a resource group. For example, you might use tags to differentiate production from development or user acceptance testing (UAT) resources. The tag applies only to the resource group and not to the resources inside the group. Think of the tag as a label you add to the box, not to the contents of the box. However, the resources in the resource group can have their own tags.

Azure Resource Manager

Azure Resource Manager (ARM) is the service that enables you to manage resources, serving as the deployment and management service for Azure. ARM is not a tool or interface. Rather, as a service it functions as the broker between management tools like the Azure portal and resource providers. For example, when you create a VM in the Azure portal, the portal sends the properties to the ARM application programming interface (API). ARM then communicates with the resource provider to create the VM.

ARM supports the use of templates for creating, managing, and monitoring resources. ARM templates are JavaScript Object Notation (JSON) files that define one or more resources and their properties to deploy a tenant, management group, subscription, or resource group. So, you can automate the deployment of an entire Azure environment by using templates. ARM templates significantly simplify the creation of resources because you only need to declare in the template what you want to create and what its properties will be, and ARM then passes that information to the Azure providers, which then actually create the resources.

Azure Subscriptions and Billing Scope

As explored in Chapter 1, "Cloud Concepts," Azure is a consumption-based service where you pay only for those Azure services that you use. This section explores the different ways to pay for Azure services and how to manage costs, billing, and related topics.

Azure Subscriptions

Just as a resource group serves as a logical container for resources, an Azure subscription serves as a logical container for your Azure resources but at a higher level. Think of a subscription as a big box that contains all your resource group boxes. Also, a resource group can only exist in one subscription. Using the box analogy, imagine you have two subscription boxes side by side. A resource group box could only exist inside one of them; it cannot be in two places at once.

In a simple Azure environment, you will likely have a single Azure subscription that contains all your Azure resources. As the complexity of your environment grows, however, you might want to use multiple Azure subscriptions to simplify resource management, billing, and cost containment.

Azure subscriptions serve multiple purposes. First, a subscription is a legal agreement associated with a specific Azure offer, each with its own rate plan, terms and conditions, and benefits. For example, a free trial offer is an example of an Azure offering that would be tied to a specific Azure subscription. If you choose an Azure free trial offer, that choice creates an Azure subscription tied to that trial offer.

Azure subscriptions also represent a payment agreement. If you choose a pay-as-you-go offer tied to a credit card, for example, the resulting subscription includes the credit card number and related billing information required to bill your Azure consumption each month to that credit card.

Subscriptions also serve as scale boundaries for Azure, imposing limits for scaling Azure resources. For example, there is a limit to the number of VMs you can create in each subscription.

Lastly, Azure subscriptions can serve as administrative boundaries, enabling you to control security, resource administration, and policies. For example, you might have administrative permissions in a subscription managed by your business group but not in a subscription managed by a different business group. Or, perhaps you manage all the VMs for your organization as part of the server team but members of the data team manage the SQL Server resources. Your server resources would be in one subscription, and the data resources would be in another. Subscriptions therefore provide a good means for segregating resources and administrative responsibilities between different business units, technical silos, and so forth.

Subscriptions are not the only mechanism to segregate administrative responsibilities in Azure. For example, you can use resource groups with different role-based access (RBAC) permissions on those resource groups to control administrative permissions on the resources they contain.

Many factors drive your decisions on how many subscriptions to have, how to manage the resources in them, whether to move resources among subscriptions, and so on. Some of these factors are explained in more detail in other chapters of this book where appropriate.

Azure Billing Accounts

A *billing account* is a mechanism that you use to pay for Azure services. You manage invoices and payments with your billing account, as well as track costs. Azure currently supports the following types of Azure billing accounts:

- **Microsoft Online Services Program:** This type of billing account is added when you sign up for Azure services through the Azure website, such as an Azure Free Account, an account with pay-as-you-go billing, or through a Visual Studio subscription.
- **Enterprise agreement:** An enterprise agreement (EA) enables you to purchase software and services from Microsoft under a (typically) multiyear agreement. For example, your organization might purchase on-premises licenses, Office 365 licenses, Azure services, and Unified Support services through your organization's EA for a given three-year period. An annual true-up reconciles the licenses that you have used and adjusts the next year's cost accordingly.
- **Microsoft Customer Agreement (MCA):** An MCA consolidates monthly Azure, Azure Marketplace, and Microsoft AppSource invoices. In some regions, an MCA is created automatically when you subscribe to pay-as-you-go or Azure free subscriptions.

You can have more than one Azure billing account. For example, you might have an MCA with a corresponding billing account, and then your company signs an EA, which

would potentially give you another billing account. This can become significant (and confusing) because the different types of billing accounts have different scope, as explained in the next section.

A fourth type of billing account for Microsoft Partner Agreements specific to cloud solution providers exists but is not covered in this book.

Billing Scope

An Azure billing *scope* is a node within a billing account that enables you to manage invoices, payments, accounts, and other Azure billing-related data. As such, billing scope helps you manage Azure costs and billing in a relatively granular way.

Azure billing scopes are not directly evaluated on the AZ-900 exam, but understanding billing scope will help you better understand subscriptions, which are covered on the exam.

Each of the three types of billing accounts offers different scope options. Table 2.1 describes the scopes for the Microsoft Online Services Program; Table 2.2 describes the scopes for enterprise agreements; and Table 2.3 describes the scopes for Microsoft Customer Agreements.

TABLE 2.1 Billing scopes for Microsoft Online Services Program

Scope	Definition
Billing account	This node represents a single administrator account for one or more Azure subscriptions.
Subscription	A subscription is a grouping of Azure resources, and the Subscription scope enables invoices to be generated for each subscription, each with its unique associated payment methods.

TABLE 2.2 Billing scopes for enterprise agreements

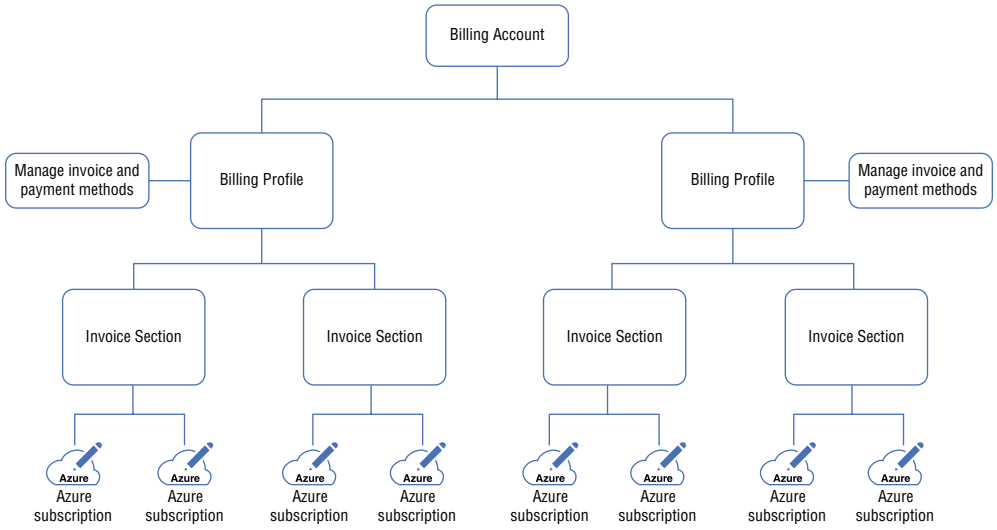
Scope	Definition
Billing account	This node represents a single administrator account for one or more Azure subscriptions.
Department	This scope is an optional grouping of enrollment accounts.
Enrollment account	This scope represents a single account owner under which subscriptions are created.

TABLE 2.3 Billing scopes for Microsoft Customer Agreements

Scope	Definition
Billing account	This node represents a customer agreement for various Microsoft services and products. The billing account is then structured using the billing profile and invoice sections, which are described later.
Billing profile	This scope represents an invoice and its corresponding payment methods. Invoices are generated at the billing profile scope, but the invoice can include multiple invoice sections.
Invoice section	This scope represents a group of costs within an invoice, with subscriptions and other costs associated with specific invoice sections.

It's perhaps easiest to visualize billing scopes with an illustration. Figure 2.4 illustrates the scope relationships within a billing account for a Microsoft Customer Agreement.

FIGURE 2.4 Billing scopes for a Microsoft Customer Agreement account



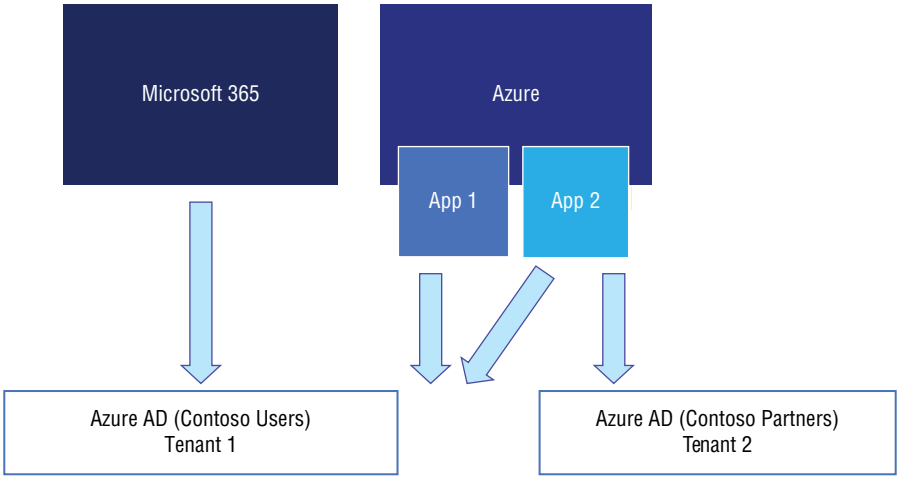
The most important thing to understand about billing scopes at this point is that they make it possible for you to more effectively manage billing and cost management within your Azure environment.

Azure Tenants

Azure tenancy bears some discussion when considering Azure billing and subscriptions. An Azure *tenant* is a specific instance of Azure Active Directory (AAD) that contains accounts and groups. In simpler terms, a tenant is a group of users. The tenant provides authentication services for your cloud resources, whether solely or in concert with onsite AD services.

Your Azure implementation is not limited to a single tenant. A *multitenancy* implementation is one in which more than one tenant share Azure resources. Figure 2.5 illustrates an example of a multitenant app deployed in Azure.

FIGURE 2.5 A shared application is an example of multitenancy.



As you might have guessed, you can use a single tenant for multiple Microsoft cloud offerings. For example, you might deploy Microsoft 365, Dynamics 365, and Azure resources all associated with a single AAD tenant. Or, you might have multiple tenants in different scenarios. It all depends on your organization's needs.

Core Azure Services

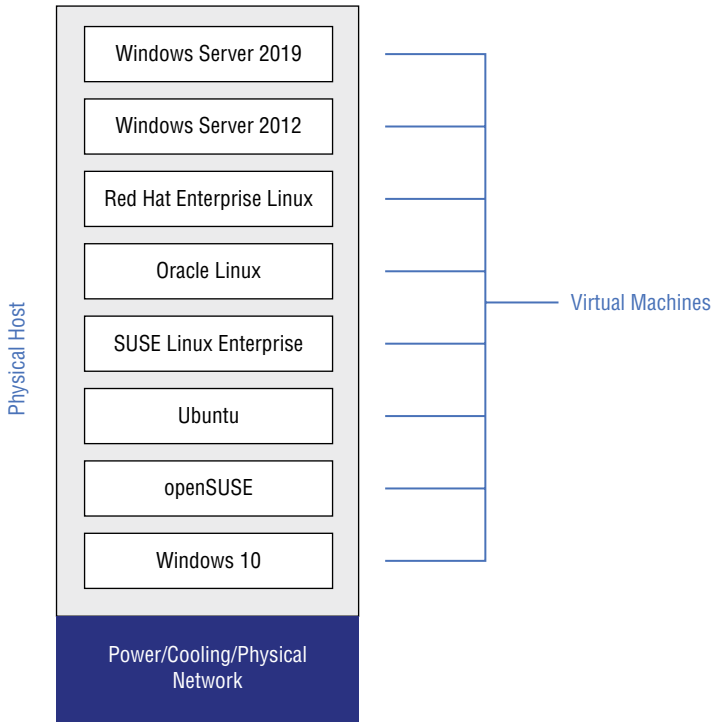
Now that you have some understanding of how Azure is structured in terms of resource management, you are ready to begin learning about the core services available within Azure. The following sections explore these core services.

Virtual Machines

In the early days of computing, systems were always physical, meaning that a computer was a discrete collection of hardware that ran an operating system and often used various input and output devices. Imagine a data center where every computer is a separate, physical device. You can also imagine that the requirements for power, cooling, networking, and management would be significant.

Virtualization changed that model. Although physical computers are still a major component of many organizations' IT infrastructure, organizations are increasingly moving toward the use of *virtual machines* (VMs). A virtual machine is an emulation of a computer system that provides the functionality of a physical computer. The VM still runs on a physical device generally referred to as a *host*. The VM is a *guest* on that host. Software called a *hypervisor* manages the VMs running on the host. Figure 2.6 illustrates a virtual host with multiple guests.

FIGURE 2.6 Multiple guest VMs on a physical host



Virtual machines offer many advantages over physical machines. First, they reduce power and physical space requirements because multiple VMs can run on a single host. Granted, the physical host must have sufficient physical CPUs, memory, and other hardware to host

the VMs, but they nevertheless host virtual systems more efficiently. For example, the data center rack space needed to host 10 physical servers might host 100 or more virtual servers. There is no specific ratio of physical-to-virtual machines because the requirements of each VM might vary. However, you can see that virtualization offers significant potential for reducing data center footprint.

VMs also offer considerable flexibility over physical machines. Because VMs are software based, you can create them very quickly by specifying their parameters and handing the actual creation task over to the hypervisor. A physical server that might take a day or more to deploy and get operational could take just minutes to deploy virtually.

Scalability is another advantage to using VMs. You can easily add resources like CPUs, cores, and memory to a VM by modifying its configuration settings. When you need to add another server for horizontal scaling, you can do so in a matter of minutes, either by initiating the process yourself or by allowing a preconfigured virtual machine scale set (described in the next section) to scale out automatically.

VMs are not limited to running whatever operating system is deployed on the host. A physical host might be running Windows Server, for example, but you might have several VMs on that server running Windows 10, Windows Server, and a selection of various Linux distributions.

Finally, VMs can be moved easily from one host to another because they are software based. You only need to move the metadata that defines the VM, not a physical server. For example, assume you need to move a VM from one region to another to accommodate a reorganization. The best method to achieve the move is to configure site recovery with the target region, move the VM using site recovery, and then fail over from the original VM to the new one.

The Azure Site Recovery service enables organizations to easily replicate virtual machines and physical servers from a primary site to a secondary site in support of business continuity. Not only can you replicate VMs between regions, but you can also replicate physical servers or VMs from your own data center to Azure.

Virtual Machine Scale Sets

A virtual machine scale set simplifies creating and managing a group of load-balanced VMs. A scale set can automatically scale out or scale in to adjust to changes in demand. Load balancing adjusts automatically to ensure that the access to the VMs in the set is balanced across VMs appropriately as VMs are added or removed from the set. Scale sets therefore enable high availability for your VMs and the services that rely on them. Scale sets also make it very simple to manage many VMs.

A scale set supports up to 1,000 standard VM instances, or up to 600 instances if you create and upload your own custom images.

The VMs in a scale set are all created from the same OS image, which ensures consistency across all VMs in the scale set. So, if a VM in the scale set hangs or is removed through a scale-in event, the remaining VMs can continue to service requests. If you need to include applications and other required components in the VMs, you create an instance with the appropriate services and configuration, save the image, and then use that image to create the scale set. All VMs in the set will then have the same components, applications, and configuration.

Azure scale sets can use either Azure Load Balancer or Azure Application Gateway to balance traffic to the VMs in the set. Which one you choose depends on the requirements of the app hosted on the VM.

Although scale sets offer high availability in themselves, you can also use availability zones to further improve availability by distributing the VMs across multiple data centers. If a data center experiences an outage, the VMs in other data centers can continue to function and service application requests.

Availability Sets

Availability sets are another feature of Azure that help you avoid potential outages caused by hardware issues, updates, or other events. Two elements that enable availability sets are *update domains* and *fault domains*. A fault domain is a logical grouping of hardware that shares a power source and network switch, similar to a physical rack in a data center. An update domain is a logical group of hardware that undergoes maintenance activities or reboot events at the same time.

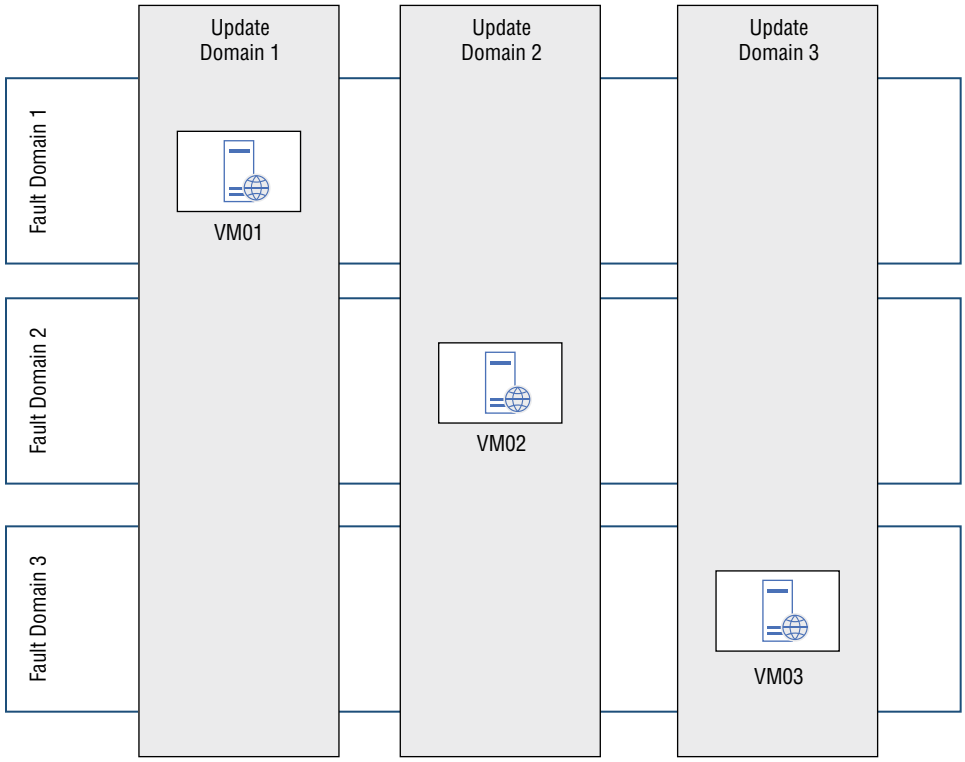
An availability set distributes VMs across multiple fault domains and update domains (see Figure 2.7). Distributing the VMs in this way helps guard against outages caused by a power or networking event in a fault domain and also enables the VMs to be updated or otherwise maintained within their respective update domains without causing the set as a whole to be unavailable.

The use of availability zones affects your SLA. As of this writing, the following scenarios are available:

- 99.99 percent: All virtual machines with two or more instances deployed across two or more availability zones in the same Azure region will have connectivity to at least one instance 99.99 percent of the time in a specified SLA period.
- 99.95 percent: All virtual machines with two or more instances deployed in the same availability set will have connectivity to at least one instance 99.95 percent of the time in a specified SLA period.
- 99.9 percent: Any single instance virtual machine using Premium SSD or Ultra Disk for all operating system and data disks will have connectivity to that instance 99.9 percent of the time in a specified SLA period.

- 99.5 percent: Any single instance virtual machine using Standard SSD Managed Disks for all operating system and data disks will have connectivity to that instance 99.5 percent of the time in a specified SLA period.
- 95 percent: Any single instance virtual machine using Standard HDD Managed Disks for operating system and data disks will have connectivity to that instance 99.5 percent of the time in a specified SLA period.

FIGURE 2.7 Availability sets distribute VMs across multiple fault domains and update domains.



Based on this list, achieving 99.95 percent or higher SLA for your VMs requires the use of availability sets or availability zones. However, you can achieve 99.9 percent SLA on a single VM instance by using premium storage, which uses solid-state drive (SSD) storage on the same physical device hosting the VM.

Azure App Service

The Azure App Service is a PaaS offering that enables you to quickly develop and deploy web applications. Azure App Service supports many development languages, including .NET, Java, Ruby, and Python, among others. Web apps developed and deployed using Azure App Service run and scale on Windows and Linux environments.

The Azure App Service also enables you to quickly build and deploy mobile back ends and REST APIs, but this chapter focuses specifically on development of web applications using Azure App Service.

Azure App Service offers much more than just development tools. It encompasses load balancing, autoscaling, automated management, and security features to support not only the development of your web application, but also hosting and scaling to make it easy to deploy and manage your web applications. It also enables your web applications to scale appropriately to respond to demand changes.

Azure App Service offers many benefits to not only simplify web app deployment but also provide a broad range of options. In addition to support for multiple development languages, Azure App Service offers support for both Windows and Linux, patches and updates the operating system and language frameworks for you, supports containers and Docker, supports the scaling and high availability features of Azure, includes access to a number of application templates in the Azure Marketplace, and much more.

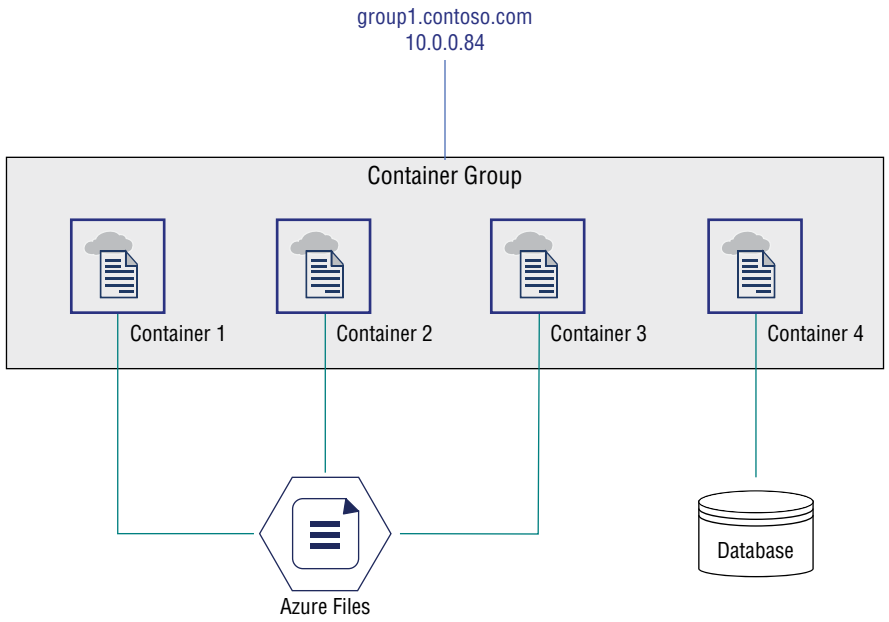
Azure Container Instances

Docker is an open source project for automating the deployment of containers, and Docker *containers* provide a means for packaging and deploying applications virtually. The container serves as a virtual environment that includes the resources necessary for its hosted application to function. For example, assume you have a simple web application that includes a web front-end server to handle web requests, an application server to handle app processing, and a database to store and serve the data needed by the other two servers. You could deploy that solution using VMs, but that requires managing the OS environments and other resources within the VMs. Containers abstract much of that and provide only the resources needed by the application so that you only need to deploy an image to the container, rather than build a VM and manage the OS and other resources. Although the container relies on the underlying OS on which it runs, you don't have to configure or manage the VM. Instead, you focus solely on the application.

Azure Container Instances (ACI) is the Azure service that gives you the ability to create and deploy containerized applications. ACI supports both Windows and Linux containers. Containers offer a potentially significant cost savings because you are only paying for consumption of CPU and memory resources used by the container, rather than paying for a VM instance. In addition, containers can easily scale out to accommodate demand changes for the containerized app.

Although ACI supports only single container instances for Windows as of this writing, ACI supports container groups for Linux. A container group is a collection of containers that run on the same host machine and share the same operating system, lifecycle, local network, resources, and storage. The group shares a single IP address and DNS name. Figure 2.8 illustrates a container group.

FIGURE 2.8 An example of a container group



ACI supports Azure Files (discussed later in this chapter) to store data used by a container, enabling the container to persist data for stateful applications.

Azure Kubernetes Service

Managing a few container instances is relatively easy, but as the number of containers increases, deployment and management become much more complex. That is where Azure Kubernetes Service (AKS) comes into play.

AKS is a container *orchestration service* that monitors container health, provides container scalability, and enables resource sharing among containers in a Kubernetes cluster. Each of the containers in the Kubernetes cluster is called a *node*. AKS simplifies deployment

because once you've defined a container image, you can use AKS to easily deploy as many instances of that image as needed within a cluster, as well as deploy multiple clusters.

Within the context of the AZ-900 exam, anywhere you see a reference to container orchestration and large-scale management of containers, it is referencing Kubernetes.

Windows Virtual Desktop

Windows Virtual Desktop (WVD) is an Azure service that enables users to run a Windows client in the cloud. The user accesses the Windows client either through a Virtual Desktop client application on their Windows device or through an HTML 5 browser like Edge or Chrome. The Windows client is available for Windows 64-bit, Windows 32-bit, and Windows ARM64 for Windows 10, Windows 10 IoT Enterprise, and Windows 7 client devices. Using a browser enables non-Windows devices, such as Android and Apple devices, to connect to and use a WVD session.

WVD offers several benefits, not least of which is the capability to run Windows on a broad range of devices running Windows, Linux, iOS, and macOS. This gives users across your organization access to Windows applications from potentially inexpensive devices. It also enables you to easily roll out line-of-business applications to your users without deploying them to individual users' devices.

WVD is also beneficial in a distributed work environment, with users often working from home or from remote offices. Rather than provision a new Windows device, install applications on it, and ship the device out to a new user, you can instead simply provision the user in Azure, and the user can then connect to a desktop session and potentially access applications in an hour or so.

Core Azure Storage

The previous sections discussed several core Azure services. None of those services could function without storage, and Azure offers several core storage offerings. This section describes these storage offerings.

Blob Storage

Azure Blob storage is optimized to store very large amounts of unstructured data such as text and binary data. For example, if you need to store many documents of various types, Blob storage is a great solution. Blob storage can be accessed in several ways, including through HTTP or HTTPS, the Azure Storage REST API, Azure PowerShell, Azure CLI, or an Azure Storage client library.

When you think of Blob storage, think of situations where you need to store either many files or large files. Examples include documents, as already discussed; video and audio files; large amounts of data to be analyzed; and backup and recovery data.

Blob Storage Tiers

How you use storage and the type of storage you choose will determine how expensive that storage will be. You should take into account how you will access data to determine which storage solution is right for each situation. For example, you might access some data frequently and other data only seldom, if ever. In this scenario you should choose the least expensive option that meets your overall needs. At the opposite end of the spectrum, you might have data that is accessed frequently and therefore requires a different, more expensive storage option.

Azure Storage provides three *access tiers* for Blob storage to enable you to not only fine-tune the type of storage you need but also minimize costs. These tiers are as follows:

- **Hot access:** This tier is optimized for storing frequently accessed data.
- **Cool access:** This tier is optimized for data that you access infrequently and for a relatively limited period of time, and that typically would move to a lower tier of storage when access is no longer likely.
- **Archive access:** This tier is intended for data that you rarely access, if at all. An example is long-term storage for backups.

Hot and cool access tiers store data online, and the archive access tier stores data offline. The hot access tier has a lower access cost than the cool access tier but also has a slightly higher SLA and higher storage cost. Choosing the cool access tier means lower storage cost but higher access cost. This is generally a sound trade-off because you should access data stored in the cool access tier less frequently. The archive access tier provides the lowest storage cost of the three but has the highest cost to rehydrate the data from offline to storage. As you begin to develop a storage hierarchy and overall plan, keep these trade-offs in mind.

Disk Storage

Azure disks are virtualized storage presented as a disk and attached to a virtual machine, much like a physical disk in a server. Azure disks are designed for 99.999 percent availability through replicas, with a disk encompassing three replicas of the data. Even if two replicas fail, the third enables the disk to persist and the data to remain available.

Azure offers three main disk roles: data disk, OS disk, and temporary disk. OS and data disks are persistent, meaning they don't go away if you reboot a VM or redeploy it. Temporary disks are not managed and do not *necessarily* persist during maintenance events and reboots, although data stored on a temporary disk will persist during a normal, successful reboot of the VM that hosts it. Temporary disks therefore should be used only for swap files, temporary files, and other data that could be lost.

Azure supports direct upload of local virtual hard disks (VHDs) to Azure managed disks, making it relatively easy to move on-premises VMs to Azure and to move or back up large data disks to Azure.

Azure managed disks support two types of encryption: server-side encryption and disk encryption. Server-side encryption provides encryption-at-rest to safeguard your data and meet compliance and policy requirements. Server-side encryption is enabled by default for all managed disks, snapshots, and images. Disk encryption on Windows volumes uses BitLocker, and Linux volumes use DM-Crypt. Disk encryption enables you to encrypt OS and data disks.

File Storage

Azure Files is another storage type supported by Azure. Think of Azure Files as files that are available securely from anywhere in the world but not associated with a specific VM or volume letter. Files stored using this service can be accessed by using the Server Message Block (SMB) protocol or Network File System (NFS) protocol. Azure file shares can be concurrently accessed by on-premises as well as Azure services.

Azure Files support many file sharing scenarios, including replacing your existing on-premises file servers or NFS file shares, moving data from on-premises to Azure, and sharing application settings or other files that your applications need to access. You can also use Azure Files to store persistent data that enables you to build stateful containers, where the container instances access the shared file system at startup.

The types of storage discussed in this section are only some of the storage types available in Azure, but these are the core storage services used by multiple Azure services.

Storage Accounts

Before you can begin using storage in Azure, you must create a *storage account*. A storage account contains Azure Storage objects and provides a unique namespace through which you can access those storage objects via HTTP and HTTPS. As you might expect, Azure offers several types of storage accounts, each intended for specific purposes and each with different costs. The following list provides a brief overview of each type of storage account:

- **General-purpose v1:** This is a legacy account type intended for blobs, files, queues, and tables.
- **General-purpose v2:** This storage account type is also intended for blobs, files, queues, and tables, as well as Data Lake Gen2.
- **BlockBlobStorage:** This storage account type is intended for block blobs and append blobs in high-performance scenarios such as high storage transaction rates, or where storage consists of small objects and/or low latency.

- **FileStorage:** This storage account type is intended for files-only storage scenarios where premium performance is required.
- **BlobStorage:** This is a legacy blob-only storage account type.

Creation and use of storage accounts is more complex than what is described here, but for the purposes of the AZ-900 exam, understand that there are various types of storage accounts suited to specific uses and you must create a storage account before you can create and use Azure storage.

Core Data Services

Azure is not just about VMs, storage, and applications. In reality, data is always the most important aspect of any IT system. There are exceptions but the role of infrastructure, storage, and applications is almost always to host and present data. As you might expect, Azure offers many types of data services.

Structured and Unstructured Data

Understanding Azure data services requires a brief discussion of *structured* and *unstructured* data. Structured data is defined by a schema that determines the characteristics and types of data stored in a data set. A table in a relational database like SQL Server is an example of structured data. The table includes columns that define which discrete data items are stored in the table, including the name of the column, its data type (like text, number, date, and so on), allowed length of the data, and other properties. Each row in the table defines a specific record or collection of columns that describe a discrete data item.

Unstructured data, as its name implies, does not have a defined structure and is not organized in a predefined way. Unstructured data often encompasses text—potentially large amounts of it—but can also contain other types of data, such as numbers and dates. Past studies have determined that the majority of data owned and managed by most organizations is unstructured data. To illustrate, consider the data that resides in all the emails, documents, spreadsheets, instant messages, and other data items you touch daily. In general, these data items represent unstructured data.

Semi-structured data is a hybrid of structured and unstructured data. Semi-structured data is not constrained to a data model like a relational data set containing tables of data, but the data does contain tags or markers that describe and enforce a hierarchy of records and fields within the data.

Azure SQL Database

To host your own database in SQL Server on-premises, you generally must deploy a server (physical or virtual), install Microsoft SQL Server on it, and use that server application to

create and manage a SQL database. Standing up and managing that server and application can be time consuming and expensive.

Azure SQL Database abstracts all the infrastructure needed to host a SQL database. It is a PaaS offering in which Microsoft hosts the SQL platform and manages maintenance like upgrades and patching, monitoring, and all other activities needed to ensure a 99.99 percent uptime for your SQL databases. The only task you focus on is creating the SQL database and managing the tables, views, and other elements within the database. So, if your organization hosts SQL Server today, moving your databases to Azure SQL Database will eliminate a significant amount of management overhead and cost.

Microsoft rolls out new features to Azure SQL Database before rolling those same features out to SQL Server. Azure SQL Database therefore gives you the best means of testing and deploying new SQL features. Best of all, you don't have to deploy servers or applications to do so.

SQL Managed Instance

SQL Managed Instance offers many of the same benefits as the Azure SQL Database service. Like Azure SQL Database, SQL Managed Instance is a PaaS offering that provides a scalable cloud data service without the need to deploy or manage hardware or SQL. Instead, you focus specifically on managing your databases.

Although these two offerings are similar, SQL Managed Instance offers additional features for auditing, authentication, backups, change data capture (CDC), common language runtime (CLR), linked servers, OPENQUERY, and several other features. A key differentiator for SQL Managed Instance is its integration with the Azure Data Migration Service, enabling organizations to easily move existing on-premises SQL instances into Azure Managed Instance.

Cosmos DB

Azure Cosmos DB is a multimodel database service that enables you to scale data out to multiple Azure regions across the world. Scaling out your data in this way makes it readily available to your users worldwide, with response times in milliseconds. Cosmos DB also provides excellent elasticity in both throughput and storage, adjusting to changes in data storage requirements and high usage during peak hours.

Cosmos DB supports SQL for querying data stored in Cosmos, but also supports other APIs, including Cassandra, MongoDB, Gremlin, and Azure Table Storage, all of which are NoSQL solutions. A NoSQL database provides the mechanism needed to store and retrieve data in nonrelational databases.

A key advantage to using Azure Cosmos DB is its support for the Gremlin API. The Gremlin API enables you to use Azure Cosmos DB to store and query massive graphs at any

scale, including those with potentially billions of vertices and edges. For context, a graph database uses a graph data type to store sets of vertices and ordered or unordered pairing of the vertices to model complex relationships. Azure Cosmos DB provides for a highly scalable solution to build and query graph-based data solutions.

Azure Database for MySQL

MySQL is an open source relational database management system that supports Structured Query Language (SQL). You can think of MySQL as an open source alternative to Microsoft's proprietary implementation of SQL.

Azure Database for MySQL gives you the capability to deploy, manage, and use MySQL databases without deploying MySQL on a server or VM. Instead, as with Azure SQL Database—which enables you to focus specifically on a SQL database rather than its infrastructure—Azure Database for MySQL lets you focus on your MySQL databases without worrying about the underlying infrastructure. Azure Database for MySQL is therefore appropriate in situations where you need to deploy and manage MySQL databases without having to manage the server, application, or other resources.

If you are familiar with web development, you might be familiar with LAMP, which stands for Linux, Apache, MySQL, and PHP. This is a common technology stack that organizations use to create web applications and websites. If you are considering moving from on-premises servers to Azure for your web applications and websites and use the LAMP stack for development, you should consider Azure Database for MySQL as an option. In the context of the AZ-900 exam, if you see a reference to LAMP, think MySQL.

Azure Database for PostgreSQL

PostgreSQL is an open source relational database management system with its origins in POSTGRES, which was a successor to the Ingress database developed at the University of California, Berkeley. Azure Database for PostgreSQL is an Azure-based implementation of PostgreSQL that supports the PostgreSQL database engine with the scalability, elasticity, high availability, and other cloud features you would expect from an Azure service. PostgreSQL is appropriate in situations where you want to deploy and manage PostgreSQL databases without worrying about underlying infrastructure.

Azure Database Migration Service

Azure Database Migration Service supports a variety of database migration scenarios for both one-time (*offline*) and continuous synchronization (*online*) migrations. In an offline migration, the source is offline while the migration takes place, making the application(s) supported by that data unavailable. In an online migration, the data is synchronized from the live source to the target and then the application is cut over to the new instance of the database.

Microsoft Marketplace

Microsoft's cloud offers incredibly rich environments with a very diverse user base. Like many Microsoft offerings, third-party providers supplement Microsoft's cloud offerings. These offerings are available through the Microsoft Marketplace, which encompasses two online stores: Microsoft AppSource and Azure Marketplace.

Microsoft AppStore offers business solutions for Azure, Dynamics 365, Microsoft 365, web apps, and Power Platform. The apps are available through the Microsoft Appstore site at `https://appsource.microsoft.com`, and through in-product experiences in M365, D365, and Power Platform.

Azure Marketplace is an online store that enables you to find and purchase a variety of Azure solutions and managed services. Think of it like Google Play in the Android world or the App Store for Apple devices. You search for apps, content, and so on in those stores on your mobile device, the content that you purchase is downloaded to your device, and you pay for it on your next phone bill.

Azure Marketplace lets you access Azure services that are built on or designed to work with Azure, as well as Azure managed services and consulting services (services delivered by third parties to help you deploy and manage Azure solutions). The resources that you obtain through Azure Marketplace are billed through your Azure account.

Summary

The services described in this chapter form the core services offered in Azure. Understanding the role these core services play will help you begin to understand how to achieve some of your initial goals in Azure and also to begin to understand some of the relationships between the various services in Azure.

This chapter covered the following concepts:

- **Core Azure Architectural Components:** Azure is a global cloud offering deployed to various geographies across the globe. Within those geographies (markets) are multiple regions, with regions paired with other regions for availability, fault tolerance, and redundancy. Availability zones ensure those same characteristics within regions.
- **Azure Subscriptions and Billing Scope:** Azure billing is tied to an Azure account, with three types of billing accounts available, including enterprise agreement, Microsoft Online Services Program, and Microsoft Customer Agreement. Your Azure subscriptions are associated with an Azure billing account. Subscriptions act as containers for Azure resource groups and, along with billing accounts, enable you to control how your Azure services are billed and paid for.

- **Core Azure Services:** Azure includes several core compute services, including virtual machines (VMs), which can leverage virtual machine scale sets and availability sets for scalability and high availability. Azure App Service makes it easy to build and deploy web applications without worrying about the underlying infrastructure needed for development or deployment. Container instances are another example of a core Azure service. Containers enable you to easily roll out and manage VM instances tailored for specific uses. Azure Kubernetes is a container orchestration service that you use to monitor and manage large numbers of container instances.
- **Core Azure Storage:** Supporting most Azure services in some way are the various storage services available in Azure. Blob storage is used for storing large amounts of data for access in various ways, disk storage is used primarily for virtual disks for VMs, and file storage lets you store files independently of a logical disk.
- **Core Data Services:** Azure includes many types of data management services for structured, semi-structured, and unstructured data. These data services encompass relational database solutions such as SQL Server for storing tabular data and NoSQL solutions such as Cosmos DB for storing and managing unstructured and semi-structured data.

Exam Essentials

Describe the core Azure architectural components. Geographies align to markets but can generally be considered as aligning to countries or regions. For example, Europe is a geography that encompasses multiple countries. The United States is another geography that represents a single country. Within each geography are regions. Within the regions, availability zones enable services to be distributed across multiple physical data centers to ensure high availability, resiliency, and fault tolerance. An availability zone has its own power, cooling, and network resources, so that if an incident occurs in one availability zone, that incident does not affect resources in other availability zones.

Resource groups are logical containers that you use to group together Azure resources and enable you to control access to the resources and their management, and otherwise manage the resources in the group as a whole. The service that enables management of resources is Azure Resource Manager (ARM). ARM lets you create resources in a declarative way using templates, which it then passes to the target Azure service provider to create the service.

Billing is another key concept in Azure. Azure billing accounts are the mechanism by which you are billed for Azure services. Subscriptions serve as a container for Azure resources, and a resource can exist only in one subscription, although you can move resources between subscriptions. Subscriptions serve as a billing boundary, enabling you to charge different groups within your organization for various Azure resources.

Describe some of the core products available in Azure. Although certainly not the only Azure service, virtual machines (VMs) are in some ways one of the most easily understood and certainly one of the most common Azure products. A VM is a virtual instance of a computer running as a guest operating system on a physical host device. A given host can run both Windows and Linux guests.

Virtual machine scale sets enable you to easily scale VMs out or in as needed to accommodate demand changes. The scale set includes load balancing to distribute load among the VMs in the set. Because all of the VMs in a scale set are based on the same image, scale sets also make it easy to roll out many VMs at once.

An availability set distributes VMs across multiple fault domains and update domains. Distributing the VMs in this way helps guard against outages caused by power or networking events in a fault domain, and it also enables the VMs to be updated or otherwise maintained within their respective update domains without causing the set as a whole to be unavailable.

Azure App Service is a PaaS offering that simplifies developing and deploying web applications. The Azure App Service takes care of the underlying infrastructure so that you can focus on developing the app, and the service also provides for deployment, load balancing, and other resources necessary to deploy and manage your app.

The Azure Container Instance service supports the creation and management of containers, which is a virtualized environment that includes the resources necessary for its hosted application to function.

Review Questions

1. Is the underlined portion of the following statement true, or does it need to be replaced with one of the other fragments that appear below?

   An Azure geography always corresponds to a specific country.

   A. corresponds to a single country or a market encompassing multiple countries.

   B. determines where your resources can reside.

   C. represents physical data centers.

   D. No change is needed.

2. An Azure region _____. (Choose all that apply.)

   A. corresponds to a specific data center.

   B. is paired with another region to help ensure high availability.

   C. can span multiple countries.

   D. specifies the location of Azure resources.

3. Is the underlined portion of the following statement true, or does it need to be replaced with one of the other fragments that appear below?

   Azure China is only available to Chinese government entities.

   A. has less restrictive regulations than other Azure geographies.

   B. is a physically isolated instance of Azure.

   C. includes Azure services that are only available in China.

   D. No change is needed.

4. An Azure region _____.

   A. specifies the location of Azure resources.

   B. is always paired with another region.

   C. contains one or more data centers.

   D. All of the above.

5. You need to deploy three virtual machines that will host an application. You want the VMs to reside in the same region, but you want to guard against power or other potential outages. You also need to ensure minimum latency between the instances. Which option describes a scenario that meets your requirements and is the most cost effective?

   A. You deploy an additional set of three VMs to a different region and use continual replication between the two regions, then fail over to the other region in the event of an outage.

   B. You place the VMs in separate resource groups in the same region.

   C. You use separate availability zones for the VMs.

   D. You use separate availability sets for the VMs.

6. Which of the following correctly describes an availability set? (Choose all that apply.)
   - **A.** Protects against power, cooling, or other physical outages but requires distribution of additional instances to other availability zones to enable rolling updates.
   - **B.** Two or more VM instances deployed to the same availability set results in a 99.99 percent SLA.
   - **C.** Distributes VM instances across multiple fault and update domains to guard against outages caused by a data center outage and to enable VMs to be updated without making all instances in the set unavailable.
   - **D.** A and B.

7. Is the underlined portion of the following statement true, or does it need to be replaced with one of the other fragments that appear below?

   A resource group in Azure serves as a logical container for Azure resources.
   - **A.** provides high availability for resources within the resource group.
   - **B.** protects resources in the group from being deleted.
   - **C.** contains resources only from the region in which the resource group resides.
   - **D.** No change is needed.

8. Is the underlined portion of the following statement true, or does it need to be replaced with one of the other fragments that appear below?

   Applying a tag to a resource group propagates the tag to all resources contained in the group.
   - **A.** prevents resources in the resource group from being deleted if the tag is a CanNotDelete tag.
   - **B.** determines the actions that administrators can take on resources in the group.
   - **C.** applies the tag only at the container level.
   - **D.** No change is needed.

9. You are planning a deployment of resources in Azure of various types to support a new project, and you want to use templates to simplify deployment and ensure that the new resources are configured the same as your existing Azure resources. Which one of the following would you use?
   - **A.** Resource groups
   - **B.** Azure Resource Manager
   - **C.** Azure Resource Templates
   - **D.** None of the above

10. Your CIO suggests the possibility of moving some of your organization's resources to Azure to cut costs and improve availability and DR options. She asks you to explain how Azure subscriptions work. Choose all answers that are correct.
    - **A.** An organization can have multiple Azure subscriptions associated with either the same or different Azure AD tenants.
    - **B.** A subscription can contain resources only from a single region.

C. You can use Azure multiple subscriptions to distribute costs to multiple groups within your organization.

D. A subscription can be moved to a new Azure AD tenant.

11. As a consequence of organizational changes that require restructuring some of your IT infrastructure, you need to move virtual machines from one region to another. Which of the following methods presents the easiest solution?

A. You back up the VM, restore it to the new region, and delete the original VM.

B. You move the VM to a resource group located in the new region.

C. You use Azure Resource Manager to move the resource to the new region.

D. You configure site recovery between the regions, migrate the VM to the new region using site recovery, and fail over to the new VM.

12. Which of the following is *not* a feature of Azure App Service?

A. Support for multiple development languages, including Java and Python

B. Support for Windows and Linux

C. Firewall protection for apps you develop with Azure App Service

D. Support for containers

13. You decide to use Azure Container Instances (ACI) to deploy containers as part of a project to deploy a new solution. You need to describe the benefits of using containers to your project team. Which of the following does not describe containers in Azure?

A. Containers can run on either Windows or Linux.

B. Containers represent a single application and the application's dependencies.

C. All containers in a container group share the same operating system.

D. Containers require setup and configuration of a virtual machine hosting them.

14. You need to deploy a stateful application using Azure Container Instances. Which of the following provides storage, enabling the application to store and retrieve persistent state?

A. Azure Disk

B. Azure Files

C. Azure Blob

D. Azure Archive

15. Your organization is planning to deploy a containerized solution in Azure and needs a container orchestration service that enables you to coordinate application upgrades and easily scale out containers. Which solution meets these goals?

A. Deploy the solution using Azure Container Instances (ACI).

B. Deploy containers using scale sets.

C. Deploy the containers using the Docker Management Portal (DMP).

D. Deploy the solution using Azure Kubernetes Service (AKS).

16. You want to deploy a solution that uses SQL to store and retrieve data on sales managers, sales quotas, and seller attainment. You want to minimize cost and configuration effort. Which solution achieves these goals?
    - **A.** Use Cosmos DB to host the data.
    - **B.** Use Azure SQL Database to host the data.
    - **C.** Use a VM with SQL Server installed to host the data.
    - **D.** None of the above.

17. Which of the following Azure services is designed for storing nonstructured data and includes support for NoSQL?
    - **A.** Azure SQL Database
    - **B.** Azure HDInsight
    - **C.** Azure Database for MySQL
    - **D.** Azure Cosmos DB

18. Your organization needs to provide a consistent user experience for running Windows applications across your enterprise, including for macOS, iOS, and Android devices. Which of the following Azure resources provide that consistent experience?
    - **A.** Azure Client Emulator
    - **B.** Windows 10 Enterprise
    - **C.** Windows Virtual Desktop
    - **D.** Microsoft 365

19. You have been tasked by your CIO with moving a large amount data from on-premises to Azure. The data needs to be maintained for compliance reasons but will not be accessed unless required by an audit or litigation. Which type of storage is the most cost effective?
    - **A.** Cool access storage
    - **B.** File storage
    - **C.** Disk storage
    - **D.** Archive access storage

20. You have set up a new Azure subscription and need to deploy storage to support a virtual machine. What is the first thing you must do to add storage?
    - **A.** Enable the subscription to support storage.
    - **B.** Create a storage account.
    - **C.** Choose the appropriate blob storage tier.
    - **D.** Nothing, because the VM includes blob hot access tier storage by default.

21. You are considering migrating several SQL instances from on-premises to Azure. Which of the following PaaS solutions provides the best support for SQL and the easiest migration path?

    A. Azure Database for MySQL

    B. Azure SQL Database

    C. Azure SQL Database Premium

    D. SQL Managed Instance

22. You are a project manager for a project to move several key web applications from on-premises to Azure. The development team has stated that support for LAMP is critical to simplifying the development effort. Which of the following is the most likely choice to support LAMP development?

    A. Azure Database for MySQL

    B. Azure SQL Database

    C. Azure Cosmos DB

    D. Azure Database for PostgreSQL